

CONFLICT RESOLUTION STRATEGY BASED ON DEEP REINFORCEMENT LEARNING FOR AIR TRAFFIC MANAGEMENT

Dong SUI , Chenyu MA  , Jintao DONG 

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China


Article History:

received 15 July 2022

accepted 27 December 2022

Abstract. With the continuous increase in flight flows, the flight conflict risk in the airspace has increased. Aiming at the problem of conflict resolution in actual operation, this paper proposes a tactical conflict resolution strategy based on Deep Reinforcement Learning. The process of the controllers resolving conflicts is modelled as the Markov Decision Process. The Deep Q Network algorithm trains the agent and obtains the resolution strategy. The agent uses the command of altitude adjustment, speed adjustment, or heading adjustment to resolve a conflict, and the design of the reward function fully considers the air traffic control regulations. Finally, simulation experiments were performed to verify the feasibility of the strategy given by the conflict resolution model, and the experimental results were statistically analyzed. The results show that the conflict resolution strategy based on Deep Reinforcement Learning closely reflected actual operations regarding flight safety and conflict resolution rules.

Keywords: conflict resolution, deep reinforcement learning, air traffic control, air traffic management, decision support technology, aviation.

 Corresponding author. E-mail: machenyu@nuaa.edu.cn

Introduction

In recent years, civil aviation in China has been in a stage of rapid development. In 2019, China's civil aviation industry completed 4.9662 million flights, an increase of 5.8% over 2018 (Civil Aviation Administration of China, 2021). Although the development of the civil aviation industry has slowed down due to the impact of COVID-19, the impact is temporary, and the continued traffic growth in the future will still lead to more and more flight conflicts. Therefore, a safe, efficient, real-time and intelligent conflict resolution decision-making technology should be developed for air traffic controllers (ATCOs) and manage the cognitive workload of ATCOs (Loft et al., 2007; Sperandio, 1971). In ICAO Doc. 9854 (International Civil Aviation Organization, 2005), conflict management is applied in three layers: strategic conflict management, separation provision, and collision avoidance. This paper mainly focuses on the second layer and studies the method of tactical conflict resolution to assist ATCOs in decision-making.

Traditional conflict resolution methods include mathematical programming methods, swarm intelligence optimisation or search methods, optimal control and geometric optimisation methods. Mathematical programming methods establish conflict resolution models using approaches such as mixed-integer linear programming or mixed-inte-

ger nonlinear programming (Cafieri & Omhene, 2017; Cai & Zhang, 2019; Omer, 2015; Çeçen & Cetek, 2020; Hong et al., 2017). The quality of solutions to these methods highly depends on how accurately the model portrays the conflict resolution process. However, the complex aircraft motion and the variable airspace environment are difficult to portray accurately. The swarm intelligence optimisation or search methods accurately portray the conflict resolution process by portraying the airspace operating environment through simulation, but the computing speed is slow. Swarm intelligence optimisation methods iterate to find better solutions using algorithms such as the Particle Swarm Optimization (Emami & Derakhshan, 2014) and the Genetic Algorithm (Durand et al., 1996; Ma et al., 2014). Search methods include the Monte Carlo tree search algorithm (Sui & Zhang, 2022). Optimal control methods are described below. Soler et al. (2016) regarded fuel optimal conflict-free trajectory planning as a hybrid optimal control problem. Matsuno et al. (2016) proposed a stochastic optimal approximation algorithm based on the polynomial chaos kriging method to resolve conflicts in real time. Chen et al. (2016) used a three-degree-of-freedom nonlinear point-mass model to describe the multiple aircraft conflict. The conflict resolution solution derived by the optimal control method is generally a trajectory consisting of the consecutive positions of the aircraft, which differs

significantly from the scheme of ATCOs. The geometric optimisation methods are currently the most applied in engineering. The methods use geometric analysis and theoretical derivation based on information such as the aircraft's current position and velocity vector (Gilles et al., 2001; Carreno, 2002).

These aforementioned methods are successful, but in complex scenarios, it may take hundreds of seconds to solve the model (Wang et al., 2019), which will seriously reduce the timeliness of the solution. With the development of artificial intelligence in recent years, scholars have began to use new algorithms, such as Deep Reinforcement Learning (DRL) to resolve conflicts. According to the object of auxiliary decision-making, the current research can be divided into two categories: assisting ATCOs' decision-making and aircraft-assisted decision-making. The research on assisting ATCOs' decision-making is as follows. Pham et al. (2019a, 2019b) considered the uncertainty in the real environment and used the Deep Deterministic Policy Gradient (DDPG) to train agents. They used Dog Leg manoeuvres to resolve conflicts. Tran et al. (2020) added a similarity index to the reward function to measure the similarity between the conflict resolution schemes given by ATCOs and the agent. Through this kind of reward shaping, the conflict resolution scheme given by the trained agent was similar to that given by the human controller. Based on the background of free flight, Wang et al. (2019) considered the turning radius of the aircraft and used the Actor-Critic algorithm to train agents. They used a two-dimensional heading adjustment strategy to resolve conflicts and limit the number of changes in heading angle. Based on the existing air traffic control (ATC) mode, Sui et al. (2022) used the Independent Deep Q Network to train multi-agent to solve multi-aircraft conflicts. They used speed, altitude, and heading adjustment to resolve conflicts and limit the number of conflict resolution actions. The research on aircraft-assisted decision-making includes the following. Brittain and Wei (2021, 2022) established a deep multi-agent reinforcement learning framework to maintain the autonomous interval of aircraft and ensure that different numbers of aircraft pass through an en route sector without conflict. In the paper (Brittain & Wei, 2021), their framework used PPO that incorporated a long short-term memory network to ensure the model's effectiveness when the number of aircraft changes. The authors of paper (Brittain & Wei, 2022) proposed a scalable autonomous separation assurance framework to guide aircraft flying through a high-density airspace sector. Ribeiro et al. (2020a, 2020b) considered the UAV operating environment and used the combination of the DDPG and the Modified Voltage Potential method to improve the conflict resolution ability of UAVs under high-density airspace.

The currently proposed DRL-based approach capable of assisting ATCOs with conflict resolution has the following limitations: 1) The study did not use the actual airspace environment, which differs from the actual situation (Pham et al., 2019a); 2) Less consideration of regulatory rules. The method is based on the assumption of free flight (Wang

et al., 2019). Currently, most countries still fly according to the fixed route; 3) Focusing more on the conflicting aircraft itself and ignoring the impact on the movement of the neighbouring aircraft (Pham et al., 2019a).

Given the limitations of existing methods, this paper studies the conflict resolution strategy in realistic airspace scenarios with the DRL method. The DRL environment is developed based on the Air Traffic Operations Simulation System (ATOSS), a high-fidelity airspace simulation system, to ensure that the training environment is close to the actual airspace environment. The proposed DRL method allows for the resolution of two-aircraft conflicts at the same level and across levels in the current airspace operating mode using the three daily adjustment methods currently used by controllers while considering the neighbouring aircraft. This approach is currently an assisted decision-making technique and cannot fully replace ATCOs. The controller acts as a monitor. When the model produces the correct decision, the controller lets it execute it; otherwise, the controller initiates a manual decision.

1. Conflict resolution model

A *flight conflict* in this paper is defined as a case in which the horizontal separation between two aircraft is less than 10 km and the vertical separation is less than 300 m at a particular time. The conflict resolution methods mainly include altitude adjustment, speed adjustment, and heading adjustment according to the control operation regulations (Ministry of Transport of the People's Republic of China, 2017). Altitude adjustment is 300 m, speed adjustment is 10 kt and heading adjustment using route offsets. The offset turning angle is 30 degrees, and the offset distance is 6 nm from the original route.

Referring to the review (Ribeiro et al., 2020c), *tactical conflict resolution* in this study is defined as resolving a two-aircraft conflict that exists after 5 min. Therefore, the duration of the conflict resolution process is 5 min. Assume that every 1 min is a resolution period; that is, every 1 min, one aircraft in a two-aircraft conflict can be given a command or no command. The above settings allow the issuing of commands to be discrete to match the existing ATC mode. The resolution period can also be adjusted to suit a particular ATC mode.

The airspace is considered the environment, the airspace situation is considered the state, and the time to resolve the two-aircraft conflict is set as t . The controller agent generates an action A_t (i.e. command) for conflict resolution based on the state S_t and has the aircraft execute the command. The agent obtains the state S_{t+1} after a resolution period (1 min) and is rewarded with R_{t+1} by simulation extrapolation. The subsequent interaction process is the same as this. In the two-aircraft conflict resolution process, S_{t+1} and R_{t+1} depend only on S_t and A_t and not on earlier states and actions, that is, the process satisfies the Markov property. Therefore, the conflict resolution model can be modelled as a discrete-time Markov Decision Process (MDP), and the DRL method can be used

to solve the MDP. The MDP is represented by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where \mathcal{S} is the state space, \mathcal{A} is the action space, \mathcal{P} is the state transition function, \mathcal{R} is the reward space, and γ is the discount factor. γ is described in section 2.2.

1) State space

After a two-aircraft conflict is detected, a $200 \times 200 \times 6$ km cuboid airspace centred on the conflict point (the midpoint of the line connecting the positions of the two conflicting aircraft) is used to describe the state of the current conflict scenario. Then the cuboid airspace is discretized to describe the state; that is, the large airspace is divided into 8000 small cuboids of $10 \times 10 \times 0.3$ km. According to Patera (2007), the small cuboid is equivalent to creating a small protection space for the aircraft. The spatial discretization process is shown in Figure 1.

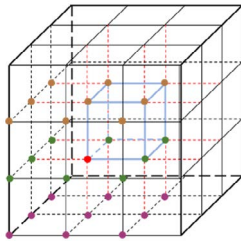


Figure 1. The spatial discretization process

During training, it is assumed that once a conflict occurs, the episode will be ended. Therefore, there is at most one aircraft in a small cuboid. In summary, the state S_t is given by Eq. (1) and can be represented as a vector:

$$S_t = (\text{Info}_1^t, \text{Info}_2^t, \dots, \text{Info}_{8000}^t), \quad (1)$$

where Info_i^t represents the airspace information of the i^{th} small cuboid at time t . If there is an aircraft in the i^{th} small cuboid at t , then:

$$\text{Info}_i^t = (\text{call_sign}, \text{type}, \text{lng}, \text{lat}, \text{alt}, \text{vspd}, \text{hspd}, \text{heading}), \quad (2)$$

where *call_sign*, *type*, *lng*, *lat*, *alt*, *vspd*, *hspd*, and *heading* are respectively the call sign, type, longitude, latitude, altitude, climb rate, horizontal speed, and heading of the aircraft in the i^{th} small cuboid. If the i^{th} small cuboid has no aircraft at t , then Info_i^t is an all-0 vector. Minimum–maximum normalization is applied to S_t .

2) Action space

The action space is the collection of the conflict resolution commands that the ATCOs can adopt in actual control operations. The commands (i.e. actions) for conflict resolution are shown in Table 1.

Table 1. The adjustment values

Resolution Method	Resolution action	Adjustment value
Altitude adjustment	Climbing or Descending, m	300, 600, 900, 1200, 1500
Speed adjustment	Acceleration or Deceleration, kt	10, 20, 30
Heading adjustment	Direct to the next waypoint Right or Left offset, nm	NULL 6

3) State transition function

Due to the complex motion of the aircraft, the MDP in this study does not have an explicit state transition function. The process of state transfer is essentially a trajectory prediction process. The aircraft trajectory predictions are calculated strictly based on the data provided by the Base of Aircraft Data (BADA) database.

4) Reward function

Optimal resolution strategy. The design of the reward function considers the following factors.

a) From the perspective of the time of conflict resolution, it should be applied as shown in Eq. (3), where i indicates that the conflict is successfully resolved at i^{th} minute, p_{time} and q_{time} are parameters, p_{time} is the maximum possible reward in the equation, and q_{time} is used to control the reward magnitude of action. The shorter the resolution time, the higher the reward for the controller agent.

$$r_1 = p_{\text{time}} - q_{\text{time}} \cdot i. \quad (3)$$

b) The action adjustment magnitude given by the model is expected to be small so that the aircraft can achieve a conflict resolution within a small magnitude. According to the adjustment method used, there are three cases.

An altitude adjustment action is rewarded in a linearly decreasing manner as shown in Eq. (4), where H_{cur} indicates the selected altitude-adjusted action value and H_{max} indicates the maximum value of the altitude adjustment action, p_{altitude} and q_{altitude} are parameters, p_{altitude} is used to control the range of maximum reward value, and q_{altitude} is used to adjust the range of reward value change. The smaller the altitude adjustment, the higher the reward for the controller agent. This is because smaller altitude adjustment can save aircraft fuel and reduce the level of discomfort for passengers.

$$r_2 = p_{\text{altitude}} - \frac{|H_{\text{cur}}|}{|H_{\text{max}}|} \times q_{\text{altitude}}. \quad (4)$$

Similarly, the reward for a speed adjustment action is shown in Eq. (5), where V_{cur} indicates the selected speed-adjusted action value and V_{max} indicates the maximum value of the speed adjustment action, p_{speed} and q_{speed} are parameters, p_{speed} is used to control the range of maximum reward value, and q_{speed} is used to adjust the range of reward value change.

$$r_3 = p_{\text{speed}} - \frac{|V_{\text{cur}}|}{|V_{\text{max}}|} \times q_{\text{speed}}. \quad (5)$$

The heading adjustment range is fixed and the reward function for a single action of heading adjustment is as shown in Eq. (6), where p_{heading} and q_{heading} are rewards for different heading adjustment actions.

$$r_4 = \begin{cases} p_{\text{heading}}, & \text{if lateral offset} \\ q_{\text{heading}}, & \text{if direct to next waypoint} \end{cases} \quad (6)$$

c) When multiple commands are given in the same scenario, the difference between the adjacent commands of the same type should be as small as possible. According to the adjustment method used, there are two cases.

The reward for the difference in altitude adjustment is as shown in Eq. (7), where H_{cur} is the value of the altitude adjustment action selected at the current time and H_{pre} is that selected at the previous moment, H_{max} is the maximum value of the altitude adjustment action, and H_{min} is the minimum value of the altitude adjustment action. p_{alt} and q_{alt} are parameters, p_{alt} is used to control the range of maximum reward value, and q_{alt} is used to adjust the range of reward value change. The smaller the adjacent altitude adjustment, the higher the reward.

$$r_5 = p_{\text{alt}} - \left| \frac{H_{\text{cur}} - H_{\text{pre}}}{H_{\text{max}} - H_{\text{min}}} \right| \times q_{\text{alt}}. \quad (7)$$

The reward for the difference in speed adjustment is as shown in Eq. (8), where V_{cur} is the value of the speed adjustment action selected at the current time and V_{pre} is that selected at the previous moment, V_{max} is the maximum value of the speed adjustment action, and V_{min} is the minimum value of the speed adjustment action, p_{spd} and q_{spd} are parameters, p_{spd} is used to control the range of maximum reward value, and q_{spd} is used to adjust the range of reward value change.

$$r_6 = p_{\text{spd}} - \left| \frac{V_{\text{cur}} - V_{\text{pre}}}{V_{\text{max}} - V_{\text{min}}} \right| \times q_{\text{spd}}. \quad (8)$$

d) If conflicts occur within the 5 min resolution time horizon, or if the two-aircraft conflict is not successfully resolved within 5 min, the reward value is as shown in Eq. (9), where p_{faile} is a negative value.

$$r_7 = p_{\text{faile}}. \quad (9)$$

In summary, the total reward value for conflict resolution is given by Eq. (10).

$$R = \left(\sum_{i=1}^7 r_i \right) / 100. \quad (10)$$

5) State termination

The controller agent takes 1 min as the resolution period to continuously generate actions to resolve conflicts until a terminal state is reached. The terminal state is reached in two conditions. One condition is that the two-aircraft conflict is resolved successfully, that is, there is no conflict between the two conflicting aircraft in the two-aircraft conflict, and there is no conflict between the two conflicting aircraft and the neighbouring aircraft within the first 5 minutes to the last 5 minutes of the two-aircraft

conflict. The other condition is the failure of conflict resolution; that is, until the last resolution period, the condition for successful conflict resolution is still not met, or there are conflicts within the 5 min resolution time horizon.

2. Model training and solving strategies

2.1. Deep Q network

The state and action data dimensions are significant in the conflict resolution problem, so the Deep Q Network (DQN) is chosen to train the controller agent. DQN combines a deep neural network with the Q-learning algorithm. The algorithm uses two key technologies: experience replay and double network structure. Figure 2 shows the principle of DQN.

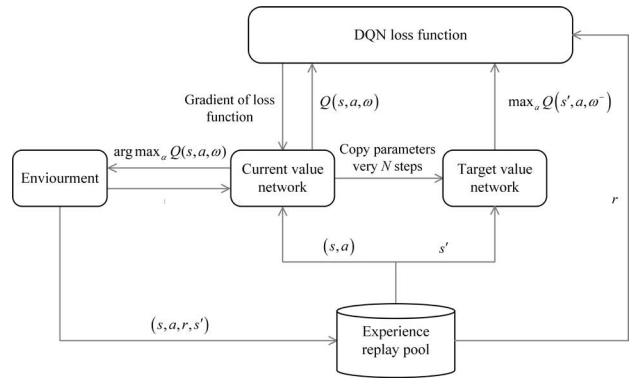


Figure 2. The principle of DQN

The experience replay solves the problems associated with sample correlation and non-static sample distribution. In the double network structure, there are two neural networks. The current value network is used to obtain the current value of Q, and the network parameters are updated during each timestep. The target value network has relatively fixed parameters and is used to obtain the Q target value, of which parameters are updated every N timesteps.

2.2. Training process

The training goal of DQN algorithm is that the loss function of the target output and the actual output error is minimized. DQN uses the target Q value calculated from the reward value and the Q value. DQN uses a Q function $Q(s, a; \omega)$ with parameters ω to approximate the value function. When the number of iterations is i , the loss function is as Eq. (11) and Eq. (12), where y_i^{DQN} indicates the target Q value, ω_i represents parameters of the current value network and ω^- represents parameters of the target value network.

$$L_i(\omega_i) = E_{(s, a, r, s')} \left[\left(y_i^{DQN} - Q(s, a; \omega_i) \right)^2 \right]; \quad (11)$$

$$y_i^{DQN} = r + \gamma \max_{a'} Q(s', a'; \omega^-). \quad (12)$$

The goal is to solve two-aircraft conflicts, it was decided to let the current value network of DQN select the aircraft performing the resolution action in the two-aircraft conflict. The structure of the current value network is shown in Figure 3. The numbers in brackets represent the number of nodes in a layer. ‘FC’ represents a fully connected layer. The network is a fully connected neural network with two hidden layers, each with 64 nodes. The dimension of the data output from the current value network is 58, twice the action space size. The current value network and the target value network have the same structure.

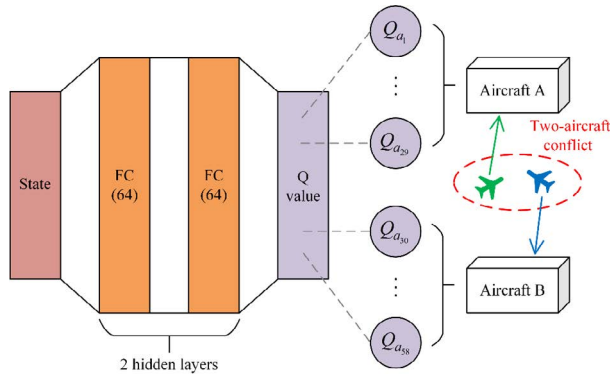


Figure 3. The structure of the current value network

Where Q_{a_1} to $Q_{a_{29}}$ are the Q value of the action corresponding to aircraft A, and $Q_{a_{30}}$ to $Q_{a_{58}}$ are the Q value of the action corresponding to aircraft B. Action a_i is the same as action a_{i+29} . DQN uses the ϵ – greedy policy to select one of the 58 Q values to select the action and the aircraft required to perform that action. According to the principle of DQN, the specific training process of the conflict resolution strategy is shown in Table 2.

3. Experiment and result analysis

3.1. Training experiment

The model was trained on a computer with 32 GB of processor RAM. The flight plan data within China’s airspace on 1 June 2018 are selected as the input, and random time changes are added to change the start times of the flight plans. A realistic airspace operating environment based on the ATOSS as the DRL environment for training the agent was developed. The ATOSS also does the state transition. Developed in our laboratory, the ATOSS combines an airspace database with a BADA-based motion simulation engine to simulate the aircraft’s operational posture. It calculates aircraft acceleration, speed, turn rate, and climb and descent rate based on parameters such as force and fuel consumption during each phase of flight. It predicts trajectories with airspace operating rules, such as aircraft altitude limits.

The ATOSS for airspace situational simulation to generate conflict scenario samples was used. During the simulation, the flight levels of the aircraft were all between 6000 m and 12000 m, with the flight level of each aircraft depending on its flight plan. When generating a conflict scenario sample, a $200 \times 200 \times 6$ km cuboid airspace in China is randomly selected, as shown in Figure 4, and the aircraft are randomly loaded in this airspace. There may be multiple conflicts in a conflict scenario sample. However, the focus was only on one of the two-aircraft conflicts (i.e. the conflict that the controller agent needs to resolve). At the time of the two-aircraft conflict in a given conflict scenario sample, there are between 10 and 30 aircraft in the cuboid airspace. The simulation generates 17865 conflict scenario samples. 1000 samples were taken for testing; the remaining samples are for training.

Table 2. DQN algorithm flow for training the agent

Algorithm: DQN algorithm for training the agent	
1:	Initialize the experience replay pool D . Initialize the action value function Q with random weight θ .
2:	Initialize the target Q network with weight $\omega^- = \omega$.
3:	Randomly select the conflict scenario, and initialize the state s_0 .
4:	Use the ϵ – greedy strategy to select the action a_t from the action space or $a_t = \arg \max_a Q(s_t, a; \omega)$.
5:	Execute command the action a_t . Receive the feedback rewards r_t and the new state of the aircraft s_{t+1} .
6:	Save the conflict sample (s_t, a_t, r_t, s_{t+1}) in the experience replay pool D .
7:	Randomly extract a conflict sample (s_j, a_j, r_j, s_{j+1}) from the experience replay pool D .
8:	If the $j + 1$ step is the final state, then $y_j = r_j$, otherwise, $y_j = r_j + \gamma \max_a Q(s_{j+1}, a'; \omega^-)$
9:	Calculate the loss function $L_t(\omega_t) = E_{(s,a,r,s')} \left[\left(y_t^{DQN} - Q(s, a; \omega_t) \right)^2 \right]$.
10:	Update $\left(y_j - Q(s_j, a_j; \omega) \right)^2$ based on the gradient descent method.
11:	Update the target Q network every 500 steps, where $\omega^- = \omega$.
12:	Loop. Until all training steps are completed.
13:	Loop. Until all training episodes are completed.

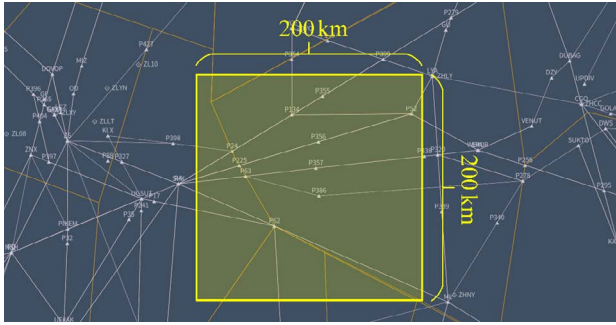


Figure 4. Schematic diagram of conflict detection simulation scenario

The hyper-parameter design of the algorithm is shown in Table 3, and the parameter values of the reward function are shown in Table 4.

1) Stability and sensitivity experiments

In this study, the average reward value and average maximum Q value are used to analyze the stability of the model. Considering that different values of parameters in Table 3 and Table 4 have different training results, in the experiment, different parameters were used for training, and each group was trained for 15000 episodes.

(a) Learning rate

As shown in Figure 5a, when the learning rate $lr = 0.001$, the reward value curve generally showed an upward trend, but the stability of the model is poor. When the learning rate $lr = 0.0005$, the convergence speed of the reward value is unchanged, and the convergence and stability of the algorithm model are better; the maximum Q value curve shows that it starts to go down and reaches convergence ahead of time, and the conflict resolution strategy score is a little higher at this time. Therefore lr is set to 0.0005.

(b) Discount factor

As shown in Figure 5b, when the discount factor is 0, the aircraft is more sensitive to short-term rewards, so its average reward value will reach the stable value earlier. As the value of the discount factor increases, the aircraft

balances the long-term and short-term rewards, so it converges slower. As observed from the maximum Q value, the model finally converges with better stability and less data fluctuation when the discount factor becomes larger. The discount factor's value does not significantly impact the final training results. However, from a theoretical point of view, we still set $\gamma = 0.99$ to ensure that the controller agent is more focused on the long-term reward, as the agent needs to resolve the two-aircraft conflict after 5 min.

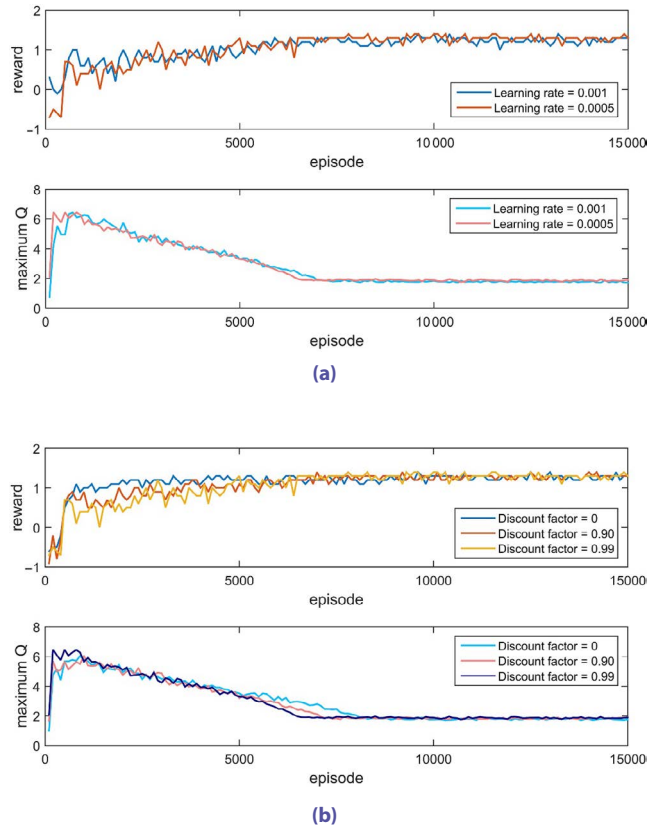


Figure 5. The training curve with different learning rate and different discount factors

Table 3. Hyper-parameters of DQN

Algorithm parameters	Parameters value	Algorithm parameters	Parameters value
Learning rate	$5e - 4$	Exploration final episode	0.02
Total training steps	30000	Discount rate	0.99
Replay pool size	50000	Target network update frequency	500
Batch size	16	Exploration fraction	0.1

Table 4. Parameter values of the reward function

Reward parameters	Value	Reward parameters	Value	Reward parameters	Value
p_{time}	120	p_{alt}	0	p_{speed}	50
q_{time}	20	q_{alt}	10	q_{speed}	60
$p_{altitude}$	50	p_{spd}	0	$p_{heading}$	25
$q_{altitude}$	50	q_{spd}	10	$q_{heading}$	15
p_{faile}	-100	-	-	-	-

(c) Reward parameters p_{fail}

As shown in Table 4, multiple parameters are involved in the reward function. Due to the lack of a unified standard for setting the reward function, these parameter values are the results of multiple tests. Many of these parameters are set to make the training result better, such as the parameters to adjust the range of altitude or speed. However, setting the reward parameter value of the conflict resolution failure is more critical in measuring the overall resolution effect. Therefore, this part takes p_{fail} as an example for test analysis, and other parameters can be tested similarly through this process. As shown in Figure 6, when the parameter value is set to -50 , -100 , and -150 , the curve of the reward value and the maximum Q value is shown in the Figure 6. It can be seen from the Figure 6 that when the value has small fluctuations, the final convergence and stability of the model are not significantly affected, indicating that the training model has a certain degree of robustness. However, when the parameter value is set to -100 , the average reward value of it is higher than the other two cases, so choose -100 as the final parameter value as shown in Table 4.

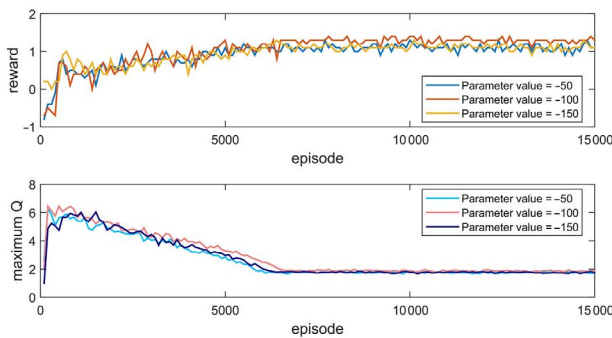


Figure 6. The training curve with different reward parameter values

Table 5. The successful resolution rate of the model trained with different steps

Training Steps	Number of successes	Successful resolution rate, %	Average reward
1,000	510	51.0	-0.854
5,000	979	97.9	1.301
10,000	992	99.2	1.375
20,000	1000	100	1.387
30,000	1000	100	1.389

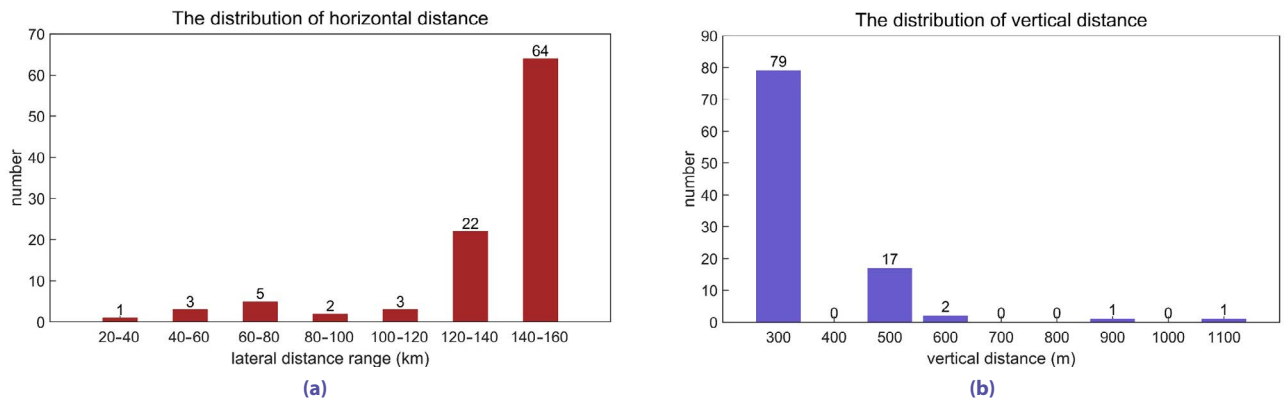


Figure 7. The distance of the closest point between the two aircraft

2) Successful resolution rate

The *successful resolution rate* refers to the proportion of the number of conflict scenario samples successfully resolved in the number of test samples. A conflict scenario sample is successfully resolved when and only when the two-aircraft conflict in this sample is successfully resolved (see 'State termination' in section 1). This study randomly selects 1000 samples for testing and sets different training steps to train the conflict resolution strategy model. The successful resolution rate analysis of the model trained with different steps is shown in Table 5. It can be seen that the successful resolution rate gradually increases as the number of training steps increases.

3.2. Testing experiment

To verify the effectiveness of the strategy given by the conflict resolution strategy model, that is, whether it conforms to the actual air traffic control operation, a total of 100 conflict scenario samples are designed for analysis in this section. According to the Doc. 4444 (International Civil Aviation Organization, 2016), in 100 samples, there are 15 same-route conflicts, 50 opposite-route conflicts, and 35 cross-route conflicts. When using DQN, all 100 conflict scenario samples are successfully resolved, and the average computation time for a resolution strategy is 1.9549e – 3s. The specific resolution strategy is analyzed from two aspects: flight safety and the rules of conflict resolution.

1) Flight safety

Flight safety is reflected by the closest distance between two conflicting aircraft. Figure 7 shows the closest points' horizontal and vertical distance distribution in all 100 testing scenarios. In all scenarios, at least one separation of the minimum horizontal and vertical directions is to meet safety requirements, which satisfies the requirements of flight safety.

The left Figure 7a shows the distribution of horizontal distance when the vertical interval between aircraft is the smallest; that is, the aircraft is at the same flight altitude. It can be seen from the Figure 7a that aircraft meet the distance constraint of more than 10 km, and most aircraft have enough intervals. The right Figure 7b shows the distribution of vertical distance when the horizontal interval between aircraft is the smallest. As seen from the Figure 7b, most aircraft are at intervals of 300 m, which satisfies the vertical limit.

2) The rules of conflict resolution

In the actual operation, the rules for conflict resolution can be summarized as follows: under the premise of ensuring successful resolution, command actions should have good controllability, high security, and a minor adjustment range. As shown in Figure 8, the commands of altitude resolution are distributed from 300 m to 1200 m. According to the regulations for conflict resolution, the altitude resolution strategy satisfies the requirements of conflicting resolution rules.

When a speed adjustment occurs, the range is distributed from 10 kt to 30 kt. For speed adjustment, if the conflict can be resolved by taking small actions, try to adjust with more minor actions. Therefore, as shown in Figure 9a, the speed adjustment strategy satisfies conflict resolution rules. From good controllability, the offset route is better than flying directly to the next waypoint strategy. The experimental results shown in Figure 9b indicate that the model chooses the route offset strategy in most cases for heading adjustment. The route offset strategy is selected 20 times, and the flying directly to the next waypoint strategy is selected seven times. Therefore, the heading adjustment strategy also satisfies the rules of conflict resolution.

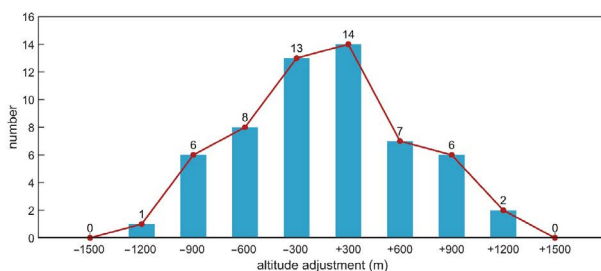


Figure 8. The distribution of the altitude adjustment

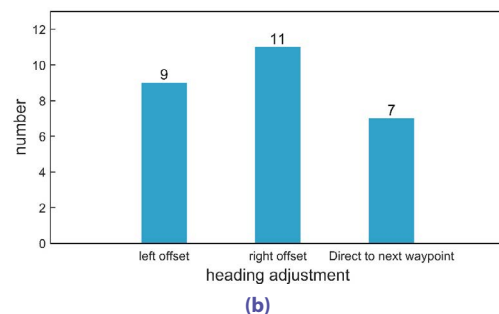
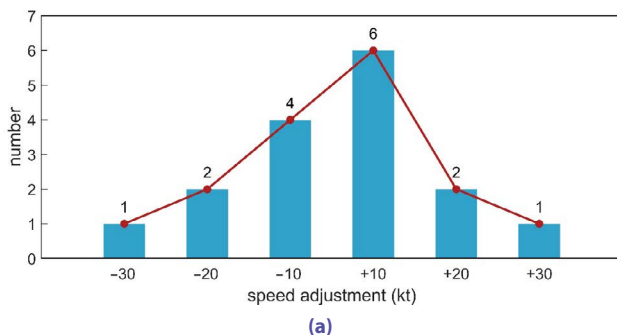


Figure 9. The distribution of the speed adjustment and the heading adjustment

Among the 100 conflict scenario samples resolved, 57 were resolved by altitude adjustment, 16 by speed adjustment and 27 by heading adjustment. The overall distribution is consistent with that of the resolution strategies used by ATCOs in actual control operations. In other words, the controller will first select the altitude adjustment, then the heading adjustment and finally the speed adjustment. However, the preference for selecting commands also varies from sector to sector and from controller to controller, and this study considers only common situations within Chinese airspace. The preference can be changed by adjusting parameters in the reward function regarding altitude, speed and heading adjustment.

3.3. Discussion

Tactical conflict resolution is an essential part of achieving intelligent ATC. In this study, the controller agent is trained using DQN to assist ATCOs in conflict resolution, and experimental proofs are performed.

The DRL environment was developed based on the ATOSS to interact with the controller agent. Realistic route structures and flight plans are used to ensure that the DRL environment approximates the actual airspace environment. Performance data for different aircraft types ensures that the solution fits the aircraft dynamics constraints. The agent uses actual controllers' height, speed and heading adjustment to resolve conflicts. The closest distance between the two aircraft involved in the conflicts is consistent with flight safety requirements. Regarding conflict resolution rules, altitude adjustments are mostly 300 m to 900 m to meet the requirement for minor adjustments. Speed adjustments are mostly 10 kt, and heading adjustments are primarily chosen as route offsets to meet the requirement of good controllability. The controller agent achieves a 100% successful resolution rate without impacting the movement of the neighbouring aircraft. In addition, DQN has a significant advantage in computation time to meet the demand for real-time resolution.

The proposed conflict resolution method still requires improvements in reliability and robustness, for example, by optimising the solution and adapting to unique scenarios. However, as the core algorithm of the decision support system for conflict resolution, there is an automation challenge: as the reliability and robustness of the algorithm

increase, the less situational awareness the ATCOs have, and the less likely they are to take over manual control if the system gives an inappropriate scheme (Endsley, 2017; O'Neill et al., 2020). This is an important issue that needs to be investigated and addressed in the future when designing conflict resolution automation.

In summary, the experimental results show that the DRL-based conflict resolution method can provide feasible suggestions for ATCOs to resolve conflicts.

Conclusions

This study establishes a two-aircraft conflict resolution model based on the MDP, and DQN is used to train the controller agent to obtain the conflict resolution strategy. Combining the experience of ATCOs, the reward function can make the behaviour of the agent after training can fit the existing ATC mode as much as possible. The testing experiments analyse the strategies the conflict resolution model gives in terms of flight safety and conflict resolution rules. It is demonstrated that the strategies given by the model are in line with the existing control regulations and the conventions of ATCOs. In addition, a successful resolution rate of 100% can be achieved, considering neighbouring aircraft. Using the DRL method reduces the time required to obtain conflict resolution strategies and can quickly deal with conflicts in the sector. It offers the possibility to reduce the workload of ATCOs and increase airspace capacity.

Acknowledgements

The research in this paper has received the help and support of the members of the Intelligent Air Traffic Control Laboratory, College of Civil Aviation, Nanjing University of Aeronautics and Astronautics.

Funding

This work was supported by the Safety Ability Project of the Civil Aviation Administration of China under Grant [number TM 2018-5-1/2]; and the Open Foundation project of The Graduate Student Innovation Base (Laboratory) of Nanjing University of Aeronautics and Astronautics under Grant [number kfjj20190720].

Author contributions

Dong Sui and Chenyu Ma conceived the study and were responsible for the design and development of the data analysis. Jintao Dong was responsible for data collection and analysis. Chenyu Ma wrote the manuscript of the paper.

Disclosure statement

All authors declare, that they have not any competing financial, professional, or personal interests from other parties.

References

- Brittain, M. W., & Wei, P. (2021). One to any: Distributed conflict resolution with deep multi-agent reinforcement learning and long short-term memory. In *AIAA Scitech 2021 Forum* (pp. 1–10). Nashville, Tennessee, USA. American Institute of Aeronautics and Astronautics. <https://doi.org/10.2514/6.2021-1952>
- Brittain, M., & Wei, P. (2022). Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 19(4), 2837–2848. <https://doi.org/10.1109/TASE.2022.3151607>
- Cafieri, S., & Omheni, R. (2017). Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes. *European Journal of Operational Research*, 260(1), 283–290. <https://doi.org/10.1016/j.ejor.2016.12.010>
- Cai, J., & Zhang, N. (2019). Mixed integer nonlinear programming for aircraft conflict avoidance by applying velocity and altitude changes. *Arabian Journal for Science and Engineering*, 44(10), 8893–8903. <https://doi.org/10.1007/s13369-019-03911-w>
- Carreno, V. (2002). Evaluation of a pair-wise conflict detection and resolution algorithm in a multiple aircraft scenario. In *NASA™-2002-211963*.
- Çeçen, R. K., & Cetek, C. (2020). Conflict-free en-route operations with horizontal resolution manoeuvres using a heuristic algorithm. *The Aeronautical Journal*, 124(1275), 767–785. <https://doi.org/10.1017/aer.2020.5>
- Chen, W., Chen, J., Shao, Z., & Biegler, L. T. (2016). Three-dimensional aircraft conflict resolution based on smoothing methods. *Journal of Guidance, Control, and Dynamics*, 39(7), 1481–1490. <https://doi.org/10.2514/1.G001726>
- Civil Aviation Administration of China. (2021). *2020 Statistical Bulletin on the Development of Civil Aviation Industry*. Beijing, China.
- Durand, N., Alliot, J., & Noailles, J. (1996, February 17–19). Automatic aircraft conflict resolution using genetic algorithms. In *Proceedings of the 1996 ACM Symposium on Applied Computing* (pp. 289–298). New York, NY, USA. American Institute of Aeronautics and Astronautics. <https://doi.org/10.1145/331119.3311195>
- Emami, H., & Derakhshan, F. (2014). Multi-agent based solution for free flight conflict detection and resolution using particle swarm optimization algorithm. *UPB Scientific Bulletin, Series C: Electrical Engineering*, 76(3), 49–64.
- Endsley, M. R. (2017). From here to autonomy: Lessons learned from human–automation research. *Human Factors*, 59(1), 5–27. <https://doi.org/10.1177/0018720816681350>
- Gilles, D., Cesar, M., & Alfons, G. (2001). Tactical conflict detection and resolution in a 3-D airspace. In *NASA CR-2001-210853*.
- Hong, Y., Choi, B., & Oh, G. (2017). Nonlinear conflict resolution and flow management using particle swarm optimization. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3378–3387. <https://doi.org/10.1109/TITS.2017.2684824>
- International Civil Aviation Organization. (2016). *Procedures for navigation services – air traffic management*. Montreal, Canada.
- International Civil Aviation Organization. (2005). *Global air traffic management operational concept*. Montreal, Canada.
- Loft, S., Neal, A., Sanderson, P., & Mooij, M. (2007). Modeling and predicting mental workload in En route air traffic control: Critical review and broader implications. *Human Factors*, 49(3), 376–399. <https://doi.org/10.1518/001872007X197017>
- Matsuno, Y., Tsuchiya, T., & Matayoshi, N. (2016). Near-optimal control for aircraft conflict resolution in the presence of uncertainty. *Journal of Guidance, Control, and Dynamics*, 39(2), 326–338. <https://doi.org/10.2514/1.G001227>

- Ma, Y., Ni, Y., & Liu, P. (2013, October 28–29). Aircrafts conflict resolution method based on ADS-B and genetic algorithm. In *Proceedings of the Sixth International Symposium on Computational Intelligence and Design* (pp. 121–124). Hangzhou, China. <https://doi.org/10.1109/ISCID.2013.144>
- Ministry of Transport of the People's Republic of China. (2017). *Air traffic management rules for civil aviation*. Beijing, China.
- Omer, J. (2015). A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers. *Computers & Operations Research*, 58, 75–86. <https://doi.org/10.1016/j.cor.2014.12.012>
- O'Neill, T., McNeese, N., Barron, A., & Schelble, B. (2020). Human–autonomy teaming: A review and analysis of the empirical literature. *Human Factors*, 64(5), 904–938. <https://doi.org/10.1177/0018720820960865>
- Patera, R. P. (2007). Space vehicle conflict-avoidance analysis. *Journal of Guidance, Control, and Dynamics*, 30(2), 492–498. <https://doi.org/10.2514/1.24067>
- Pham, D. T., Tran, N. P., Goh, S. K., Alam, S., & Duong, V. (2019a, March 20–22). Reinforcement learning for two-aircraft conflict resolution in the presence of uncertainty. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)* (pp. 1–6). IEEE. <https://doi.org/10.1109/RIVF.2019.8713624>
- Pham, D. T., Tran, N. P., Alam, S., Duong, V., & Delahaye, D. (2019b, June). A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties. In *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)* (pp. 17–21). Vienna, Austria.
- Ribeiro, M., Ellerbroek, J., & Hoekstra, J. (2020a, December). Determining optimal conflict avoidance manoeuvres at high densities with reinforcement learning. In *Proceedings of the Tenth SESAR Innovation Days* (pp. 7–10), Virtual Conference. SESAR.
- Ribeiro, M., Ellerbroek, J., & Hoekstra, J. (2020b). Improvement of conflict detection and resolution at high densities through reinforcement learning. In *Proceedings of the Conference on Research in Air Transportation (ICART)* (pp. 1–4), Virtual Conference. <http://resolver.tudelft.nl/uuid:d3bf3c0d-16bf-4ca4-b695-2868d761c129>
- Ribeiro, M., Ellerbroek, J., & Hoekstra, J. (2020c). Review of conflict resolution methods for manned and unmanned aviation. *Aerospace*, 7(6), 79. <https://doi.org/10.3390/aerospace7060079>
- Soler, M., Kamgarpour, M., Lloret, J., & Lygeros, J. (2016). A hybrid optimal control approach to fuel-efficient aircraft conflict avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 17(7), 1826–1838. <https://doi.org/10.1109/TITS.2015.2510824>
- Sperandio, J. C. (1971). Variation of operator's strategies and regulating effects on workload. *Ergonomics*, 14(5), 571–577. <https://doi.org/10.1080/00140137108931277>
- Sui, D., Xu, W., & Zhang, K. (2022). Study on the resolution of multi-aircraft flight conflicts based on an IDQN. *Chinese Journal of Aeronautics*, 35(2), 195–213. <https://doi.org/10.1016/j.cja.2021.03.015>
- Sui, D., & Zhang, K. (2022). A tactical conflict detection and resolution method for en route conflicts in trajectory-based operations. *Journal of Advanced Transportation*, 2022, 1–16. <https://doi.org/10.1155/2022/9283143>
- Tran, P. N., Pham, D. T., Goh, S. K., Alam, S., & Duong, V. (2020). An interactive conflict solver for learning air traffic conflict resolutions. *Journal of Aerospace Information Systems*, 17(6), 271–277. <https://doi.org/10.2514/1.1010807>
- Wang, Z., Li, H., Wang, J., & Shen, F. (2019). Deep reinforcement learning based conflict detection and resolution in air traffic control. *IET Intelligent Transport Systems*, 13(6), 1041–1047. <https://doi.org/10.1049/iet-its.2018.5357>